1. **In Task 1, What did you set things in *settings.py* to achieve the politeness of your crawler?**

DUPEFILTER_CLASS = 'scrapy.dupefilters.BaseDupeFilter'

FEED_EXPORT_ENCODING = 'utf-8'

2. **What lexical rules did you use to extract information in Task 2? Why?**

```
# awards, lexical
patterns = [
    [{'IS_TITLE': True}, {'ORTH': 'nomination'}],
    [{'IS_TITLE': True}, {'ORTH': 'nominations'}],
    [{'ORTH': 'Golden'}, {'ORTH': 'Lion'}],
    [{'ORTH': 'Lifetime'}, {'ORTH': 'Achievement'}],
    [{'ORTH': 'Emmy'}],
    [{'ORTH': 'Oscar'}],
    [{'IS_TITLE': True, 'OP': '+'}, {'ORTH': 'Award'}],
    [{'IS_TITLE': True, 'OP': '+'}, {'ORTH': 'Awards'}]
]
```

I try to use ‘Awards’, ‘nominations’ … as the key words and extract the award names through this lexical way. Movies awards have many different names, I thought there’s no way to extract all the award names using one rule, and I just tried to include those unique names as many as possible into my extraction rules.

```
# actors, lexical
patterns = [
    [{'IS_TITLE': True, 'OP': '+'}]
]
```

The first letter of actors’ names are usually capital. So I used ‘is_title’ to identify actor names in lexical way.

```
# education, lexical
patterns = [
    [{'IS_TITLE': True, 'OP': '+'}, {'ORTH': 'College'}],
    [{'IS_TITLE': True, 'OP': '+'}, {'ORTH': 'University'}],
    [{'IS_TITLE': True, 'OP': '+'}, {'ORTH': 'School'}]
]
```

In terms of education, I thought the institution name should include ‘college’, ‘university’, or ‘school’. Also, the education institutions are capital.

```
# movies, lexical
patterns = [
    [{'IS_TITLE': True, 'OP':'+'}, {'IS_SPACE': True}, {'ORTH': '('}, {'LIKE_NUM': True}, {'ORTH': ')'}],
    [{'IS_TITLE': True, 'OP':'+'}, {'IS_LOWER': True}, {'IS_TITLE': True, 'OP':'+'}, {'IS_SPACE': True}, {'ORTH': '('}, {'L
]
```

In this case, movies’ names in a director’s biography are usually followed by a specific year. So, I tried to extract capital words that followed by year.

```
# birthplace, lexical
patterns = [
    [{'ORTH': 'born'}, {'ORTH': 'in'}, {'IS_TITLE': True}, {'IS_PUNCT': True, 'OP':'*'}, {'IS_TITLE': True, 'OP':'*'}]
]
```

The birthplace usually follows a pattern like '… was born in …'. So I tried to use this lexical pattern to extract birthplace.

3. **What syntactic rules did you use to extract information in Task 2? Why?**

```
# awards, syntactic
patterns = [
    [{'IS_TITLE': True, 'POS': 'PROPN'}, {'ORTH': 'nomination'}],
    [{'IS_TITLE': True, 'POS': 'PROPN'}, {'ORTH': 'nominations'}],
    [{'ORTH': 'Golden'}, {'ORTH': 'Lion'}],
    [{'ORTH': 'Lifetime'}, {'ORTH': 'Achievement'}],
    [{'ORTH': 'Emmy'}],
    [{'ORTH': 'Oscar'}],
    [{'IS_TITLE': True, 'OP': '+', 'POS': 'PROPN'}, {'ORTH': 'Award'}],
    [{'IS_TITLE': True, 'OP': '+', 'POS': 'PROPN'}, {'ORTH': 'Awards'}]
]
```

I mainly build the rules on the lexical one. Because I thought movie awards have so many different names and patterns. It is quite difficult to extract this information using syntactic feature. So, I just add the 'POS' to be 'PROPN' to make it more specific and clear.

```
# actors, syntactic
patterns = [
    [{'ENT_TYPE': 'PERSON', 'OP': '+'}]
]
```

I used the entity type to be PERSON to extract actors' names. And it turns out to have better performance than the lexical one.

```
# education, syntactic
patterns = [
    [{'IS_TITLE': True, 'OP': '+'}, {'ORTH': 'College', 'POS': 'PROPN'}],
    [{'IS_TITLE': True, 'OP': '+'}, {'ORTH': 'University', 'POS': 'PROPN'}],
    [{'IS_TITLE': True, 'OP': '+'}, {'ORTH': 'School', 'POS': 'PROPN'}]
]
```

Similarly, I thought the lexical rules are better for the extraction of education institutions, so I built the syntactic rules upon the lexical one and made it more specific.

```
# movies, syntactic
patterns = [
    [{'IS_TITLE': True, 'OP':'+'}, {'IS_SPACE': True}, {'ORTH': '('}, {'ENT_TYPE': 'DATE'}, {'ORTH': ')'}],
    [{'IS_TITLE': True, 'OP':'+'}, {'IS_LOWER': True}, {'IS_TITLE': True, 'OP':'+'}, {'IS_SPACE': True}, {'ORTH': '('}, {'ENT
]
```

I added some entity types like date to the rules to make it more specific. The movies names usually follow a similar pattern. <name> (year)

```
# birthplace, syntactic
patterns = [
    [{'ORTH': 'born'}, {'ORTH': 'in'}, {'ENT_TYPE': 'GPE'}, {'IS_PUNCT': True, 'OP':'*'}, {'ENT_TYPE': 'GPE', 'OP':'*'}]
]
```

The basic logic is quite similar to the lexical one. And I added some entity type to this rule.